Defining Additional Attributes of Programs

If Additional Attributes is set to "Y", a window is displayed which contains the following additional attributes for selection:

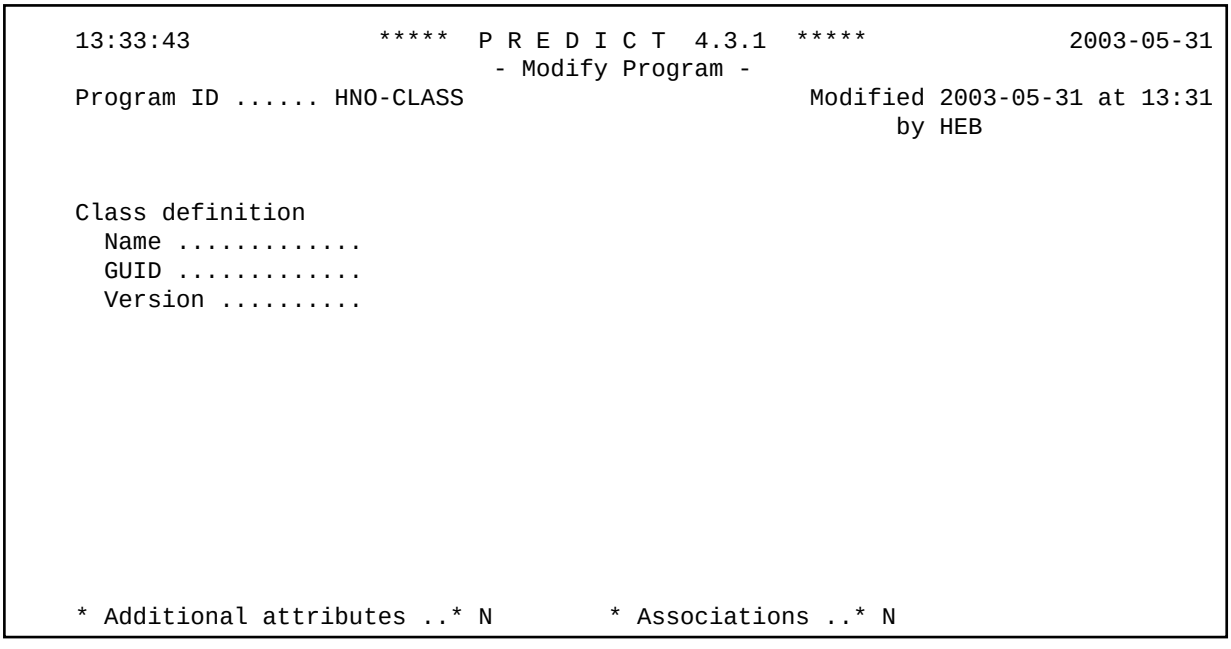
- Description
  A description of the program can be entered.
- Owner
  The owner of the program can be entered.
- Class definition
  see Programs of Type Class
- Resource definition
  see Programs of Type Resource
- Database function options
  see Programs of Type Database function
- Procedure options
  see Programs of Type Procedure
- Entry points
  Entry points are to be modified. This is valid only for programs written in certain languages. See Overview of Language-Specific Program Types.
- SQL procedure code
  Only for programs of type R and Language S.
  The SQL Procedure Editor is called.

The following rules apply:

- Only those types of additional attributes appear in the window that apply to the type of program. For example: the option Class definition is not contained in the list when a program of type Resource is processed.
- More than one choice can be made at a time. The respective input maps are then displayed one after the other.

The additional attributes are described in the following sections.

Programs of Type Class



Parameters	
Class definition	
Name	The name of the class.
GUID	The globally unique ID of the class.
Version	The version number of the class.

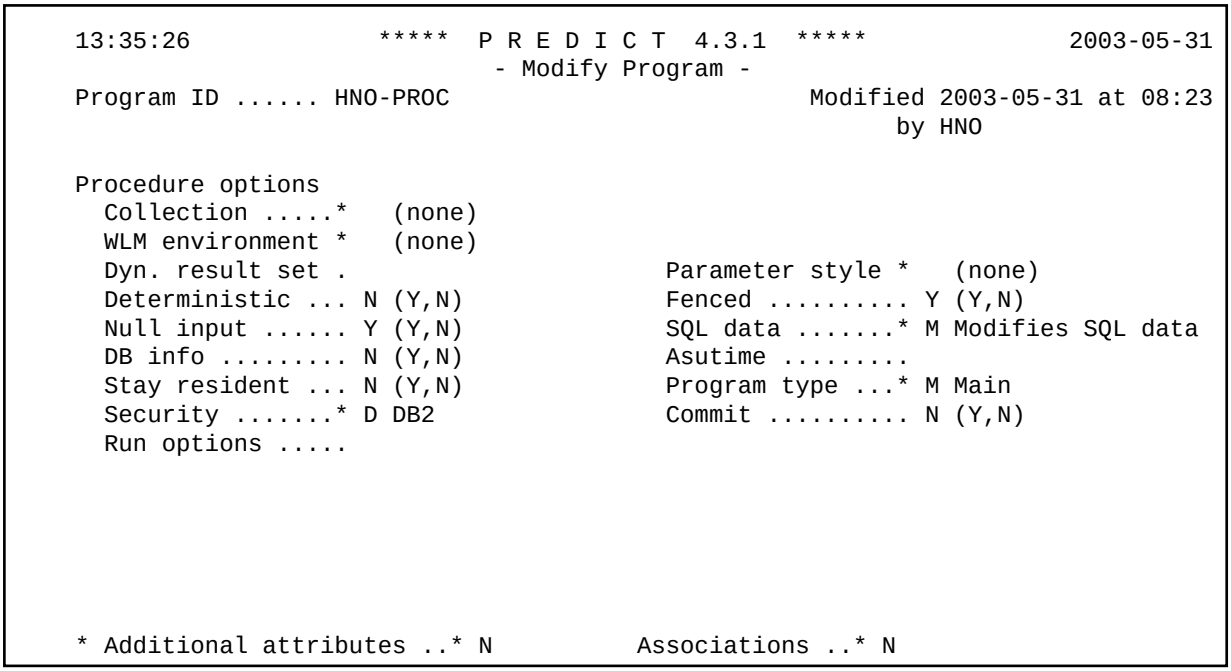
Programs of Type Resource

```
13:35:26             *****  P R E D I C T  4.3.1  *****              2003-05-31
                              - Modify Program -
Program ID ...... HNO-RES                            Modified 2003-05-31 at 08:23
                                                           by HNO


Resource definition
  File name ........




  Library ..........
  User system Fnr ..
  User system DBnr .









  Additional attributes ..* N          Associations ..* N
```

Parameters	
Resource definition	
File name	File name documented by the Predict program.
Library	The name of the library in which the file name is stored.
User System Fnr	The number of the user system file.
User System DBnr	The number of the database in which the user system file is located.

The type of Resource can be documented in the language field of a Predict program object. There is a user exit program U-PGMLAN that allows dynamic extension of possible languages in each installation.

Programs of Type Procedure



```
13:35:26            *****  P R E D I C T  4.3.1  *****            2003-05-31
                             - Modify Program -
Program ID ..... HNO-PROC                          Modified 2003-05-31 at 08:23
                                                         by HNO

Procedure options
  Collection .....*   (none)
  WLM environment *   (none)
  Dyn. result set .                        Parameter style *   (none)
  Deterministic ... N (Y,N)                Fenced ......... Y (Y,N)
  Null input ..... Y (Y,N)                 SQL data .......* M Modifies SQL data
  DB info ........ N (Y,N)                 Asutime .........
  Stay resident ... N (Y,N)                Program type ...* M Main
  Security .......* D DB2                  Commit ......... N (Y,N)
  Run options .....

* Additional attributes ..* N          Associations ..* N
```

Parameters	
Collection	Identifies the package collection. **N** NO COLLID **Y** Use collection-ID. A collection-ID must then be specified. **blank** none
WLM environment	Identifies the MVS workload manager application environment.
Dyn. result set	Specifies the maximum number of query result sets that the stored procedure can run.
Deterministic	Specifies whether the procedure returns the same results for identical arguments. **Y** Yes **N** No **blank** undefined
Null input	Specifies whether the procedure is called if any of the input arguments is null at execution time. **Y** Yes **N** No **blank** undefined

DB info	Specifies whether specific information that is included in DB2 is passed to the procedure when it is invoked. **Y** Yes **N** No **blank** undefined
Stay resident	Specifies whether the load module for the procedure remains resident in memory when the procedure ends.
Security	Specifies whether how the procedure interacts with an external security product. **D** DB2 **F** Definer **U** User **blank** none
Run options	Specifies the language environment run-time options to be used for the procedure.
Parameter style	Identifies the linkage convention use to pass parameters to the procedure. **D** DB2SQL **G** General **N** General with nulls **J** Java **blank** none
Fenced	Determines that the external procedure runs in an external address space. **Y** Yes **N** No **blank** undefined
SQL data	Indicates whether the procedure can execute any SQL statements. **M** Modifies SQL data **N** No SQL **R** Read SQL data **S** Contains SQL **blank** none
Asutime	Specifies the total amount of processor time.

Program type	Specifies whether the procedure runs as a main or a subroutine. **S** Sub **M** Main **blank** none
Commit	Indicates whether DB2 commits the transaction immediately on return from the procedure. **Y** Yes **N** No **blank** undefined

Programs of Type Database function

```
13:35:26            *****  P R E D I C T  4.3.1  *****            2003-05-31
                             - Modify Program -
Program ID ...... HNO-U-C                          Modified 2003-05-31 at 08:37
                                                         by HNO

Function options
  Function type ..*   Sourced
  Specific name ...
  Collection .....*
  WLM environment .
  Deterministic ...   (Y,N)              Fenced .........   (Y,N)
  Null input ......   (Y,N)              SQL data ......*
  External action .   (Y,N)              Scratchpad .....
  Final call ......   (Y,N)              Allow parallel .   (Y,N)
  DB info .........   (Y,N)              Cardinality ....
  Asutime .........                      Stay resident ..   (Y,N)
  Program type ...*                      Security ......*
  Run options .....




    Additional attributes ..* N          Associations ..* N
```

Parameters	
Function type	The type of the function. **S** Scalar **T** Table
Specific name	Specifies an unique name for the function.

Collection	Identifies the package collection. **N** NO COLLID **Y** Use collection-ID. A collection-ID must then be specified. **blank** none
WLM environment	Identifies the MVS workload manager application environment.
Deterministic	Specifies whether the function returns the same results for identical arguments. **Y** Yes **N** No **blank** undefined
Null input	Specifies whether the function is called if any of the input arguments is null at execution time. **Y** Yes **N** No **blank** undefined
External action	Specifies whether the function takes an action that changes the state of an object that DB2 does not manage. **Y** Yes **N** No **blank** undefined
Final call	Specifies whether final call is made to the function. **Y** Yes **N** No **blank** undefined
DB info	Specifies whether specific information that DB2 knows is passed to the function when it is invoked. **Y** Yes **N** No **blank** undefined
Asutime	Specifies the total amount of processor time.

Program type	Specifies whether the function runs as a main or a subroutine. **S** Sub **M** Main **blank** none
Run options	Specifies the language environment run-time options to be used for the function.
Fenced	Determines that the external function runs in an external address space. **Y** Yes **N** No **blank** undefined
SQL data	Indicates whether the function can execute any SQL statements. **N** No SQL **R** Read SQL data **S** Contains SQL **blank** none
Scratchpad	Specifies whether DB2 provides a scratchpad for the function
Allow parallel	Specifies whether parallelism can be used. **Y** Yes **N** No **blank** undefined
Cardinality	Specifies an estimate of the expected number of rows that the function returns.
Stay resident	Specifies whether the load module for the function remains resident in memory when the function ends.
Security	Specifies whether how the function interacts with an external security product. **D** DB2 **F** Definer **U** User **blank** none

System Programs

Programs that are only available as object code and hence have no language are documented with programs of type E (external object) and language Z (system program). Predict creates XRef data for these so called system programs because neither the preprocessor nor Natural can create XRef data for object code.

The implementation pointer for a system program has to be specified explicitly. One entry point (with the ID of the program object) is created by Predict, additional entry points have to be specified manually.

Programs of Type dynamic

Programs of type dynamic are used to document calls of programs of the same name from different steplibs depending on the library structure. The following rules apply:

- Because programs of type dynamic document any number of implemented members, no check is performed as to whether the members documented by the program are actually implemented.
- With the active retrieval function Programs using programs, programs of type dynamic are ignored as current objects.
- Programs of this type can only have children for association Uses PR concept.

